IN THE CLAIMS

1. (Currently Amended) In a digital communications network having network cards, a method comprising:

controlling applications executed within the network, wherein controlling the applications comprises,

in response to a state change message, performing a set of transitioning actions to transition~~ing~~ each of the applications between one of a plurality of active states on an active card of the network cards and one of a plurality of standby states on a standby card of the network cards, wherein the plurality of active states comprises an active ready state, a quiescent state, and a no-provisioning state, -the set of transitioning actions including:

flushing data to a disk,

synchronizing RAM with a disk database,

synchronizing RAM with the standby card, and

building RAM from the active card,

wherein ~~substantially~~ all necessary commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready sate,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of network management requests for configuring the active network card are rejected during the no-

provisioning state; and

subsequent to the transitioning, sending a state change confirmation message.

2. (Original) The method of claim 1, wherein an application state machine controls the execution of the application.

3. (Original) The method of claim 2, further comprising:

receiving control messages from a shelf manager; and

communicating via APIs to the application, wherein the shelf manager may be located on a remote network card.

4. (Canceled)

5. (Previously Presented) The method of claim 1, wherein the standby states comprises a standby locked state.

6. (Currently Amended) In a digital communications network having network cards, a method comprising:

switching the state of an application in an active state to a standby state, comprising,

in response to a first state change message, performing a set of

transitioning actions to transition~~ing~~ the application from the active state to a quiescent state on an active card of the network cards, the transitioning actions including,

flushing data to a disk, and

synchronizing RAM with the standby card; ~~and~~

subsequent to the transitioning from the active state to the quiescent state, sending a first state change confirmation message;

in response to a second state change message, transitioning the application from the quiescent state to the standby state on a standby card of the network cards,; and

subsequent to the transitioning from the quiescent state to the standby state, sending a second state change confirmation message;

wherein ~~substantially~~ all necessary commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of the commands required by each of the applications are loaded into a memory of the standby card for executing each of the applications during the standby state.

7. (Currently Amended) In a digital communications network having network cards,

a method comprising:

upgrading code of an application in an active state on an active card of the network cards to a standby locked state on a standby card of the network cards comprising,

in response to a first state change message, performing a first set of transitioning actions to transition~~ing~~ the application from the active state to a no provisioning state, the first set of transitioning actions including,

flushing data to a disk,

wherein ~~during which~~ at least a portion of network management requests for configuring the active network care are rejected during the no provisioning state;

subsequent to the transitioning from the active state to the no provisioning state, sending a first state change confirmation message;

in response to a second state change message, performing a second set of transitioning actions to transition~~ing~~ the application from the no provisioning state to a quiescent state, the second set of transitioning actions including,

synchronizing RAM with the standby card, and

flushing data to a disk,

wherein ~~during which~~ memories of the active card and the

standby card are synchronized during the quiescent state;

~~and~~

subsequent to the transitioning from the no provisioning state to the quiescent state, sending a second state change confirmation message;

in response to a third state change message, transitioning the application from the quiescent state to the standby locked state, wherein ~~during which~~ the application is in a ready state in the standby card but does not communicate with the corresponding application of the active card; and

subsequent to the transitioning from the quiescent state to the standby locked sate, sending a third state change confirmation message.

8. (Original) The method of claim 7, wherein the standby locked state does not allow disk database access nor access to write to RAM.

9. (Original) The method of claim 7, wherein the no provisioning state does not allow access to write to a disk database.

10. (Original) The method of claim 7, wherein the quiescent state does not allow access to write to a disk database nor access to write to RAM.

11. (Currently Amended) The method of claim 7, ~~In a digital communications network having network cards, a method~~ further comprising:

upgrading code of ~~an~~ the application in ~~a~~ the standby state to ~~an~~ the active state comprising,

in response to a fourth state change message, transitioning the application from the standby state on a standby card of the network cards to a no provisioning state on an active card of the network cards; ~~and~~

subsequent to the transitioning from the standby state to the no provisioning state, sending a fourth state change confirmation message;

in response to a fifth state change message, transitioning the application from the no provisioning state to the active state wherein ~~substantially~~ all necessary commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ~~ready~~ state, and wherein at least a portion of network management requests for configuring the active network card are rejected during the no provisioning state; and

subsequent to the transitioning from the no provisioning state to the active state, sending a fifth state change confirmation message.

12. (Currently Amended) In a digital communications network having network cards, a system comprising:

means for controlling applications executed within the network, wherein the means for

controlling the applications comprises,

means for performing a set of transitioning actions, in response to a state change message, to transition each of the applications between one of a plurality of active states on an active card of the network cards and one of a plurality of standby states on a standby card of the network cards, wherein the plurality of active states comprise an active ready state, a quiescent state, and a no-provisioning state, the set of transitioning actions including,

flushing data to a disk,

synchronizing RAM with a disk database,

synchronizing RAM with the standby card, and

building RAM from the active card,

wherein ~~substantially~~ all necessary commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of network management requests for configuring the active network card are rejected ruing the no-provisioning state; and

means for sending, subsequent to the transitioning, a state change confirmation message.

13. (Original) The system of claim 12, further comprising:

means for receiving control messages from a shelf manager; and

means for communicating via APIs to the application, wherein the shelf manager may be located on a remote network card.

14. (Currently Amended) In a digital communications network having network cards, a system comprising:

means for switching the state of an application in an active state to a standby state, comprising,

means for performing a first set of transitioning actions, in response to a first state change message, to transition~~ing~~ the application from the active state to a quiescent state on an active card of the network cards, the first set of transitioning actions including,

flushing data to a disk, and

synchronizing RAM with the standby card; ~~and~~

means for sending, subsequent to the transitioning from the active state to the quiescent state, a first state change confirmation message;

means for transitioning, in response to a second state change message, the application from the quiescent state to the standby state on a standby card of the network cards; and

means for sending, subsequent to the transitioning from the quiescent state to the

standby state, a second state change confirmation message;

wherein ~~substantially~~ all necessary commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of the commands required by each of the applications are loaded into a memory of the standby card for executing each of the applications during the standby state.

15. (Currently Amended) In a digital communications network having network cards, a system comprising:

means for upgrading code of an application in an active state on an active card of the network cards to a standby locked state on a standby card of the network cards comprising,

means for performing a first set of transitioning actions, in response to a first state change message, to transition the application from the active state to a no provisioning state, the first set of transitioning actions including,

flushing data to a disk,

wherein ~~during which~~ at least a first portion of network management requests for configuring the active network card are rejected during the no provisioning state, and wherein at least a

second portion of network management requests for reading configurations of the active card are processed during the no-provisioning state;

means for sending, subsequent to the transitioning from the active state to the no provisioning state, a first state change confirmation message;

means for performing a second set of transitioning actions, in response to the second state change message, to transition the application from the no provisioning state to a quiescent state, the second set of transitioning actions including,

synchronizing RAM with the standby card, and

flushing data to a disk,

wherein ~~during which~~ memories of the active card and the standby card are synchronized during the quiescent state; ~~and~~

means for sending, subsequent to the transitioning from the no provisioning state to the quiescent state, a second state change confirmation message;

means for transitioning, in response to a third state change message, the application from the quiescent state to the standby locked state, wherein ~~during which~~ an application is in a ready state in the standby card but does not communicate with the corresponding application of the active card during the standby locked state; and ~~,~~

means for sending, subsequent to the transitioning from the quiescent state to the standby locked state, a third state change confirmation message.

16. (Currently Amended) ~~In a digital communications network having network cards~~The, ~~a~~ system of claim 14, further comprising:

means for upgrading code of ~~an~~ the application in ~~an~~ the standby state to ~~an~~ the active state comprising,

means for transitioning, in response to a fourth state change message, the application from the standby state on ~~a~~ the standby card ~~of the network cards~~ to ~~a~~ the no provisioning state on ~~an~~ the active card ~~of the network cards~~; ~~and~~

means for sending, subsequent to the transitioning from the standby state to the no provisioning state, a fourth state change confirmation message;

means for transitioning, in response to a fifth state change message, the application from the no provisioning state to the active state, wherein ~~substantially~~ all necessary commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ~~ready~~ state, and wherein at least a portion of network management requests for configuring the active network card are rejected during the no provisioning state; and

means for sending, subsequent to the transitioning from the no provisioning state to the active state, a fifth state change confirmation message.

17. (Currently Amended) A computer readable medium having stored thereon a plurality of instructions for controlling tasks performed on network cards, said plurality of instructions when executed by a computer, cause said computer to perform:

controlling applications executed within the network, wherein controlling the applications comprises,

in response to the state change message, performing a set of transitioning actions to transition~~ing~~ each of the applications between one of a plurality of active states on an active card of the network cards and one of a plurality of standby states on a standby card of the network cards, wherein the plurality of active states comprise an active ready state, a quiescent state, and a no-provisioning state, the set of transitioning actions including,

flushing data to a disk,

synchronizing RAM with a disk database,

synchronizing RAM with the standby card, and

building RAM from the active card,

wherein ~~substantially~~ all necessary commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of network management requests for configuring the active network card are rejected during the

no-provisioning state; and

subsequent to the transitioning, sending a state change confirmation message.

18. (Original) The computer-readable medium of claim 17 having stored thereon additional instructions, said additional instructions when executed by a computer, cause said computer to further perform:

receiving control messages from a shelf manager; and

means for communicating via APIs to the application, wherein the shelf manager may be located on a remote network card.

19. (Currently Amended) A computer readable medium having stored thereon a plurality of instructions for controlling tasks performed on network cards, said plurality of instructions when executed by a computer, cause said computer to perform:

switching the state of an application in an active state to a standby state, comprising,

in response to the first state change message, performing a set of transitioning actions to transition the application from the active state to a quiescent state on an active card of the network cards, the set of transitioning actions including,

flushing data to a disk, and

synchronizing RAM with the standby card; ~~and~~

subsequent to the transitioning from the active state to the quiescent state, sending a first state change confirmation message;

in response to a second state change message, transitioning the application from the quiescent state to the standby state on a standby card of the network cards~~,~~; and

subsequent to the transitioning from the quiescent state to the standby state, sending a second state change confirmation message;

wherein ~~substantially~~ all necessary commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of the commands required by each of the applications are loaded into a memory of the standby card for executing each of the applications during the standby state.

20. (Currently Amended) A computer readable medium having stored thereon a plurality of instructions for controlling tasks performed on network cards, said plurality of instructions when executed by a computer, cause said computer to perform:

upgrading code of an application in an active state to a standby locked state comprising,

in response to the first state change message, performing a first set of transitioning actions to transition the application from the active state to a no provisioning state on an active card of the network cards, the first set of transitioning actions including,

flushing data to a disk,

wherein ~~during which~~ at least a portion of network management requests for configuring the active network card are rejected during the no provisioning state;

subsequent to the transitioning from the active state to the no provisioning state, sending a first state change confirmation message;

in response to a second state change message, performing a second set of transitioning actions to transition the application from the no provisioning state to a quiescent state, the second set of transitioning actions including,

synchronizing RAM with the standby card, and

flushing data to a disk,

wherein ~~during which~~ memories of the active card and the standby card are synchronized during the quiescent state; ~~and~~

subsequent to the transitioning from the no provisioning state to the quiescent state, sending a second state change confirmation message;

in response to a third state change message, transitioning the application from the quiescent state to the standby locked state on a standby card of the network cards, wherein ~~during which~~ an application is in a ready state in the standby card but does not communicate with the corresponding application of the active card during the standby locked state; and ~~,~~

subsequent to the transitioning from the quiescent state to the standby locked

state, sending a third state change confirmation message.

21. (Currently Amended) ~~A~~ The computer readable medium of claim 20, ~~having stored thereon a plurality of instructions for controlling tasks performed on network cards~~, said plurality of instructions ~~when executed by a computer,~~ cause said computer to further perform:

upgrading code of ~~an~~ the application in ~~an~~the standby state to ~~an~~ the active state comprising,

in response to a fourth state change message, transitioning the application from the standby state on ~~a~~ the standby card ~~of the network cards~~ to ~~a~~ the no provisioning state on ~~an~~ the active card ~~of the network cards~~; ~~and~~

subsequent to the transitioning from the standby state to the no provisioning state, sending a fourth state change confirmation message;

in response to a fifth state change message, transitioning the application from the no provisioning state to the active state, wherein ~~substantially~~ all necessary commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ~~ready~~ state, and wherein at least a portion of network management requests for configuring the active network card are rejected during the no provisioning state; and

subsequent to the transitioning from the no provisioning state to the active state, sending a fifth state change confirmation message.

22. (Currently Amended) In a digital communications network, a system for controlling tasks performed on network cards comprising:

a CPU subsystem;

one or more input/output ports connected to the CPU subsystem for communicating with the network; and

special hardware connected to the CPU subsystem via a bus, wherein the CPU subsystem controls applications executed within the network, wherein the applications receive a state change message, wherein the applications performing a set of transitioning actions in response to the state change message to~~that~~ transition from one of a plurality of active states on an active card of the network cards and one of a plurality of standby states on a standby card of the network cards, wherein the applications send a state change confirmation message subsequent to the transition actions, wherein the plurality of active states comprise an active ready state, a quiescent state, and a no-provisioning state, -wherein the set of transitioning actions includes:

flushing data to a disk,

synchronizing RAM with a disk database,

synchronizing RAM with the standby card, and

building RAM from the active card,

wherein ~~substantially~~ all necessary commands required by each of the applications are loaded into a memory of the active card for executing each of the applications during the active ready state,

wherein memories of the active card and the standby card are synchronized during the quiescent state, and

wherein at least a portion of network management requests for configuring the active network card are rejected during the no-provisioning state.

23. (Original) The system of claim 22 further comprising a disk database connected to the CPU subsystem via a PCI bus.

24. (Original) The system of claim 22, wherein the CPU subsystem comprises:

a central processing unit;

a system controller connected to the central processing unit;

random access memory connected to the system controller; and

an application state machine for transitioning applications between one of a plurality of active states and one of a plurality of standby states.